REMARKS

Receipt of the Office Action of May 3, 2004 is gratefully acknowledged.

In response to paragraph 3 of the Office Action, the figures have been amended to insert proper English identifications

In response to paragraph 4 of the Action, the title of the invention has been changed which is more indicative of the invention as claimed.

In response to paragraph 5 of the Action, the abstract has been amended.

Regrading the specification, it is noted that it already contains the necessary headings suggested in the MPEP

Claims 1 - 8 have been replaced with new claims 9 - 16. New claims 9 - 16 are believed to fully comply with the provisions of 35 USC 112, first and second paragraph.

In regard to new claim 9, please note the following two points:

(1) Since the functional units (i.e. arithmetic/logic unit (80), load/store unit (83), etc.), the common data bus (8) and the like are basically the same as those for register-based superscalar processors capable of out-of-order execution and there can be various variants with different detailed structures, they are summarized as "means for executing operations involved in issued instructions out of order" in claim 9. Besides, though the data buffer in claim 9 may appear to be different from that in claim 1, this is a problem of expression. As is apparent from page 17, lines 7-16 in the specification or claim 5, this is not a change of the matter.

(2) The word "consolidated register file" is rather redundant for use in claims. So, instead, the word "register file" is used in claim 9. Besides, the words "decode" and "issue" are used in almost the same implication in the specification and in claims 1-8, though only the word "issue" is used in claim 9.

With respect to paragraphs 7 - 12 of the Action, please take notice of the

following quotation from the specification:

> Stack state such as {..., word 1, word2, word3, word4} (the right end is the top of the stack) in a traditional stack machine corresponds to state in which state of pointer stack is such as { ..., <a>, <b>, <c>, <d>} (the right end is the top of the stack), and in which word1, word2, word3 and word4 are respectively held in the entries of the consolidated register file whose addresses are <a>, <b>, <c> and <d>, in the computer system according to the present invention. (page 2, line 20 - page 3, line 2)

And, for better understanding of the matter, please refer to the example action described in the specification (page 28, line 4 - page 47, line 15).

As is apparent from the above, the invention disclosed in the specification is related to the stack machine, which has an operand stack. On the other hand, Yeager et al., US Patent 5,758,112, Walker, US Patent 5,881,305 and An Efficient Algorithm for Exploiting Multiple Arithmetic Units, Tomasulo, R. M. are related to the register machine, which has a number of architected (logical) registers. Therefore, paragraphs 13 - 30 of the Action are not conceived to apply to the present patent application.

It is important to note that an improvement according to the present invention lies in the existence and utilization of the data buffer, which can hold lower operand-stack elements that are common to advanced and completed operand stacks. This improvement is not found in the references of record.

Finally, copies of English abstracts of the prior art cited in the previously filed IDS are being submitted herewith so that this art can be considered by the examiner in the examination of this application.

In view of the foregoing, reconsideration and re-examination are respectfully requested and claims 9 1 - 16 found allowable.

Respectfully submitted,

Felix J. D'Ambrosio
Reg. No. 25,721

September 3, 2004

BACON & THOMAS, PLLC
625 Slaters Lane, 4th Floor
Alexandria, VA 22314 - 1176
(703) 683-0500
S:\Producer\fjd\CLIENTS\Kyomei Int'l Patent & TM Office\SEKI3003\Sept 3 2004 Response.wpd

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-040105
(43)Date of publication of application : 13.02.1998

(51)Int.Cl. G06F 9/38

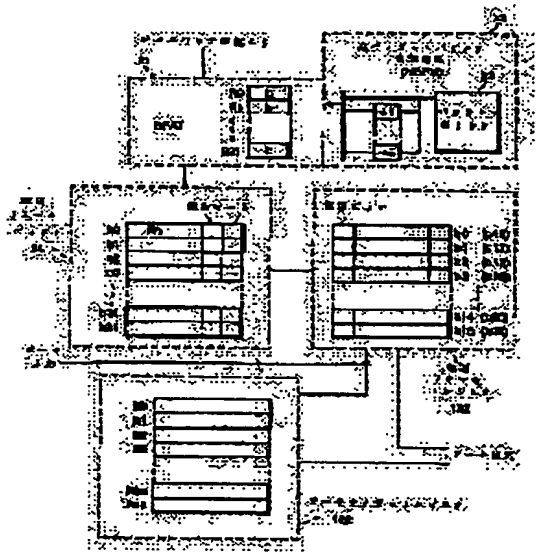
(21)Application number : 09-084803
(22)Date of filing : 03.04.1997

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>
(72)Inventor : KIN S CHAN
HAN Q RII
DAN Q NGUIEN

(30)Priority
Priority number : 96 633267 Priority date : 16.04.1996 Priority country : US

(54) METHOD AND PROCESSOR FOR ALLOCATING RENAME REGISTER

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a method and device which make good use of a virtual buffer so as to increase the parallel processing functions for instructions of a pipeline type processor.
SOLUTION: A buffer pointer allocating mechanism is configured and rename buffer are allocated even unless a physical rename register 123 is ready for use during dispatching. According to a buffer pointer allocation table 21, those virtual rename buffers 24 are allocated. When an instruction corresponding to an entry stored in a specific physical rename register 123 is completed, virtual bits set together with respective physical rename registers 123 are inverted, and the results are written in an architecture register 122. Therefore, rename registers can be allocated in dispatching as if more rename buffers than the actual physical rename registers 123 were present.



LEGAL STATUS

[Date of request for examination]	11.12.1998
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	3093673
[Date of registration]	28.07.2000
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's	

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-333905
(43)Date of publication of application : 18.12.1998

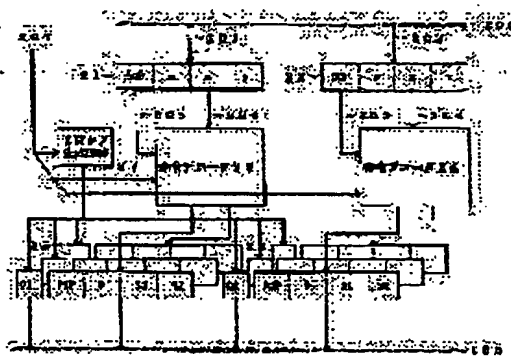
(51)Int.Cl. G06F 9/38
G06F 9/38

(21)Application number : 09-141687
(22)Date of filing : 30.05.1997

(71)Applicant : KOFU NIPPON DENKI KK
(72)Inventor : TAKATO MASAHIKO

(54) INFORMATION PROCESSOR FOR SUPER SCALAR SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To correctly control even the occurrence of even to be processed between software instructions by imparting a tag to a software instruction unit and recognizing the break of the software instruction.
SOLUTION: As for the software instructions stored in a software instruction queue, by an instruction issuance permission instruction 207, a first SW instruction is decoded by an instruction decoder 23, a second SW instruction is decoded by the instruction decoder 24 and they are developed to plural firmware instructions. At the time, the first SW instruction is developed to the two pieces of firmwares, the second SW instruction is developed to the three pieces of the firmwares and they are stored in firmware instruction queues 25 and 26. Further, an SW tag number 01 and the two firmware instructions are stored in the firmware instruction queue 25, the SW instruction tag number 02 and the three firmware instructions are stored in the firmware instruction queue 26 and the break of the software instruction is recognized.



LEGAL STATUS

[Date of request for examination]	30.05.1997
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	2901573
[Date of registration]	19.03.1999
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	19.03.2003

decision of rejection]
[Date of extinction of right] 28.07.2003

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-067878
(43)Date of publication of application : 11.03.1994

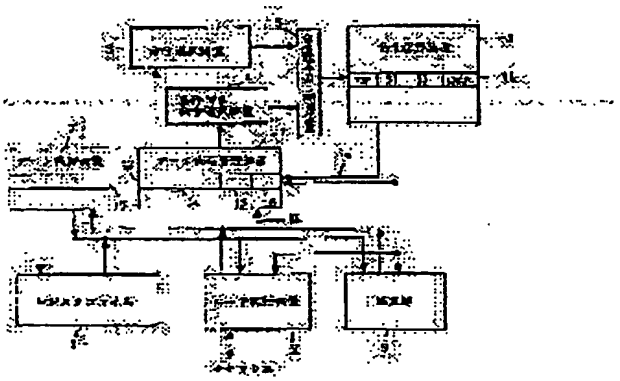
(51)Int.Cl. G06F 9/38

(21)Application number : 04-217771
(22)Date of filing : 17.08.1992

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD
(72)Inventor : MIYAKE JIRO

(54) INFORMATION PROCESSOR

(57)Abstract:
PURPOSE: To shorten a program execution time by reducing excess cycles in the case of an instruction such as a load instruction whose data acquisition is not previously known.
CONSTITUTION: This information processor is provided with an instruction storage device 1 for temporarily storing instructions, a data dependence managing device 2 for detecting instructions to be surely executed and instructions to be conditionally executed out of instructions stored in the device 1, an instruction selector 3 for selecting some instructions out of the instructions to be surely executed which are detected by the device 2, a conditional instruction selector 4 for selecting some instructions out of the instructions to be conditionally executed which are detected by the device 2, and an instruction reading control device 5 selecting some instructions out of the instructions selected by the selectors 3, 4, reading out the selected instructions from the device 1 and controlling the start of their execution.



LEGAL STATUS

[Date of request for examination]	16.02.1993
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	2125738
[Date of registration]	13.01.1997
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	31.07.2002

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-024926
(43)Date of publication of application : 29.01.1999

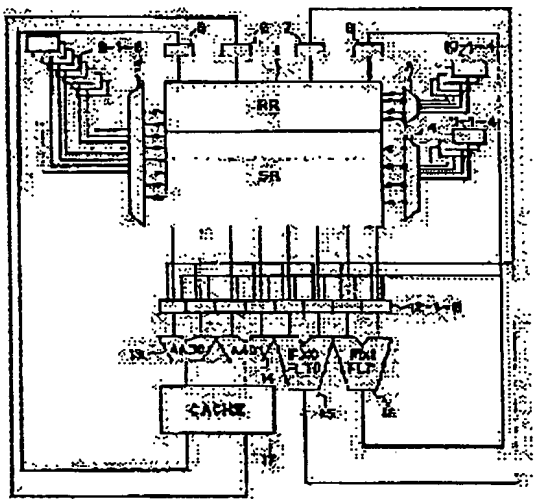
(51)Int.Cl. G06F 9/38

(21)Application number : 09-195069
(22)Date of filing : 04.07.1997

(71)Applicant : NEC CORP
(72)Inventor : NISHIDA MASATO

(54) INFORMATION PROCESSOR

(57)Abstract:
PROBLEM TO BE SOLVED: To provide register renaming control with extremely simple control and to drastically improve storage and the increase of delay by making data which are written to a general register the content of a renaming register that is read.
SOLUTION: As for read of operand data, an instruction that completes write onto a general register SR and an instruction which is not written to the register SR because a preceding instruction is not completed from the register SR, they are read from a renaming register RR. A memory configuration of RR/SR does not have to distribute the double number of bit lines nor select read data of the RR/SR any more by sharing bit lines on the side of read. Also, the width of the RR/SR is made even by making the number of write ports uniform, storage is improved and chip layout is facilitated.



LEGAL STATUS

[Date of request for examination]	04.07.1997
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	2943776
[Date of registration]	25.06.1999
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-013770
(43)Date of publication of application : 17.01.1995

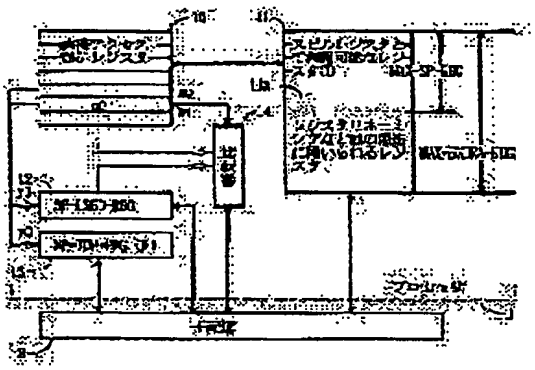
(51)Int.Cl. G06F 9/45
// G06F 9/34

(21)Application number : 05-144097
(22)Date of filing : 16.06.1993

(71)Applicant : FUJITSU LTD
(72)Inventor : KAWABA MOTOYUKI

(54) COMPILING METHOD AT THE TIME OF USING ADDITIONAL REGISTER OF LOAD STORE TYPE PROCESSOR

(57)Abstract:
PURPOSE: To improve the operating speed and the performance of a compiler by making a compiler such a constitution where the compiler recognizes such variables that had to be assigned to a main storage although the compiler could be assigned to a register, gives the priority degree for arrangement of spill registers to the variables, and assigns the variables of higher priority degree to the higher order addresses of the spill registers.
CONSTITUTION: A spill total number register 12 is provided with a head address register 13 which stores the head address B of the variable assigned to a main storage device 2, and a comparing means 14. When a compiler detects to the effect that a part of variables desirable to be assigned to a register 10 of a load store type processor 1 to which a direct access is possible could not be assigned to the register 10, the compiler performs a compiling job to selectively perform accesses to the variables assigned to the spill registers and the variables assigned previously to the storages device 2 by the setting, loading and storing instructions.



LEGAL STATUS

[Date of request for examination]	21.12.1999
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	3296027
[Date of registration]	12.04.2002
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	